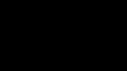


Image Capture Framework

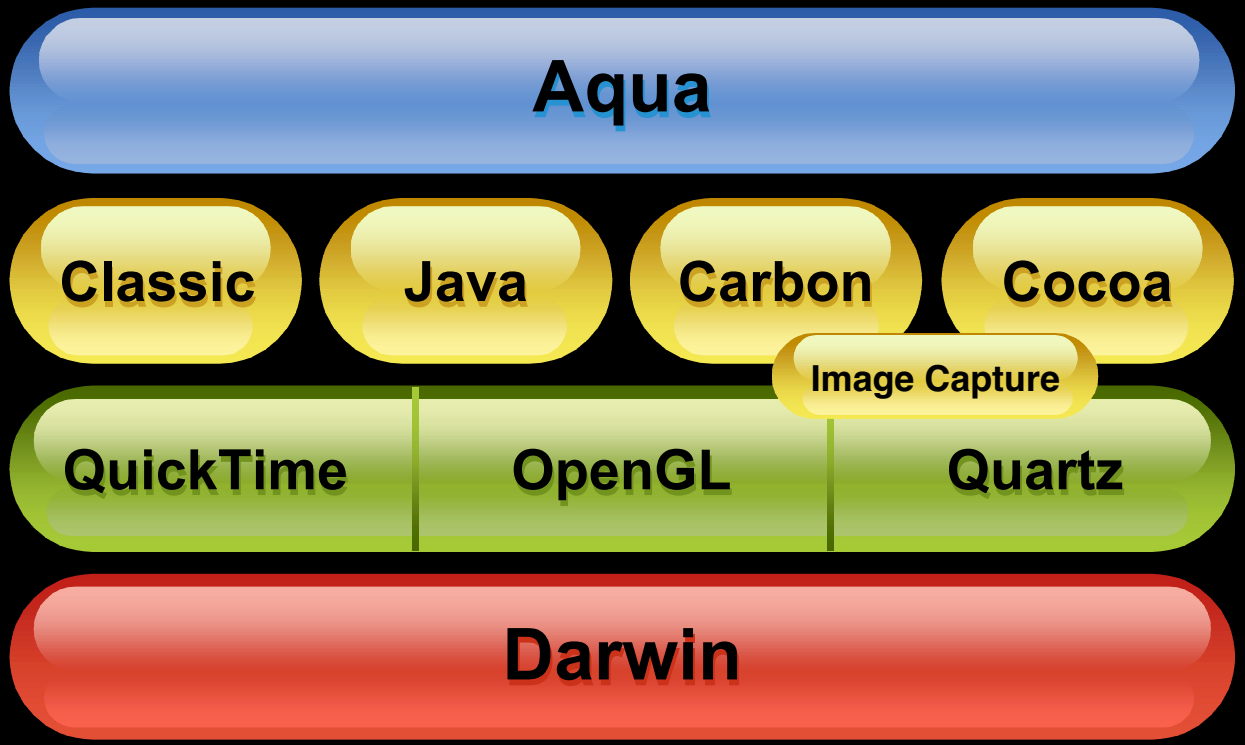
Session 515

Image Capture Framework

Werner Neubrand
Engineer

Agenda

- Image Capture Overview
- Scanner Support in Image Capture
- TWAIN Framework
- Changes in Jaguar
- Q&A


Aqua
Classic
Java
Carbon
Cocoa
Image Capture
QuickTime
OpenGL
Quartz
Darwin

What Is Image Capture?

- Image Capture Framework is a technology for working with image capture devices
  - Abstracts the device specifics from your application
  - Defines driver architecture for digitalcameras and scanners
  - Supports standard architectures and protocols

Image Capture Workflow

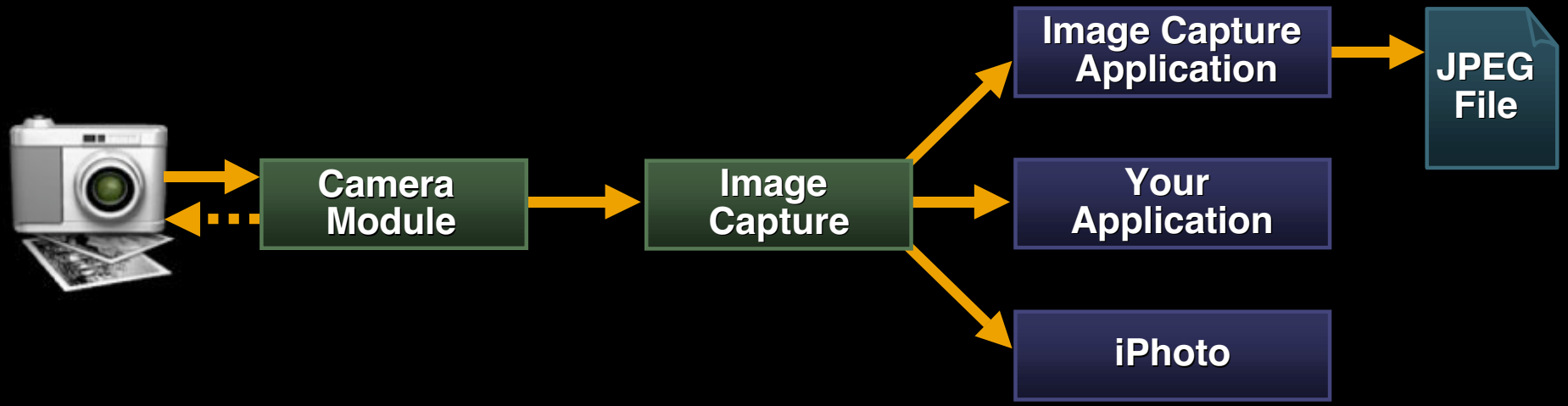
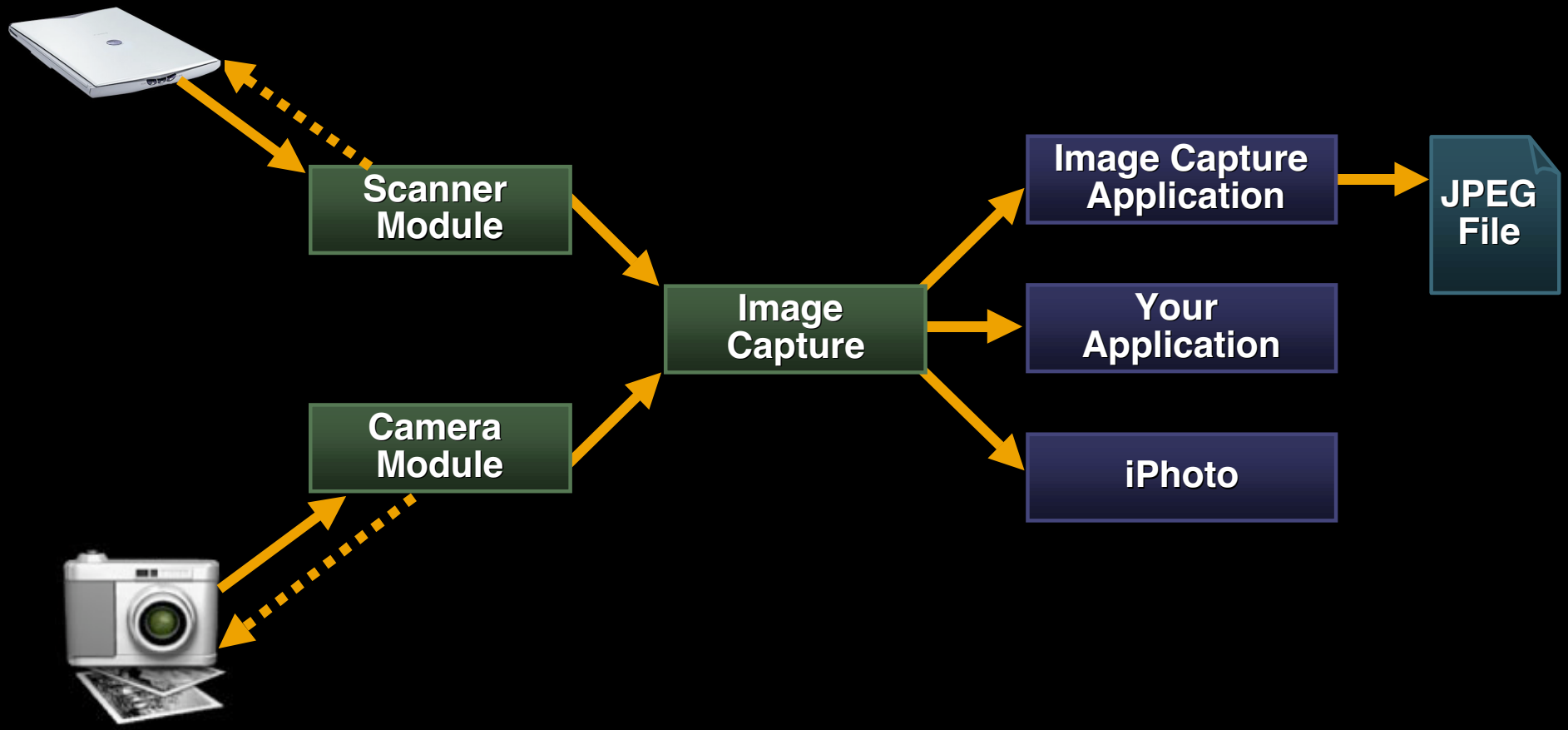
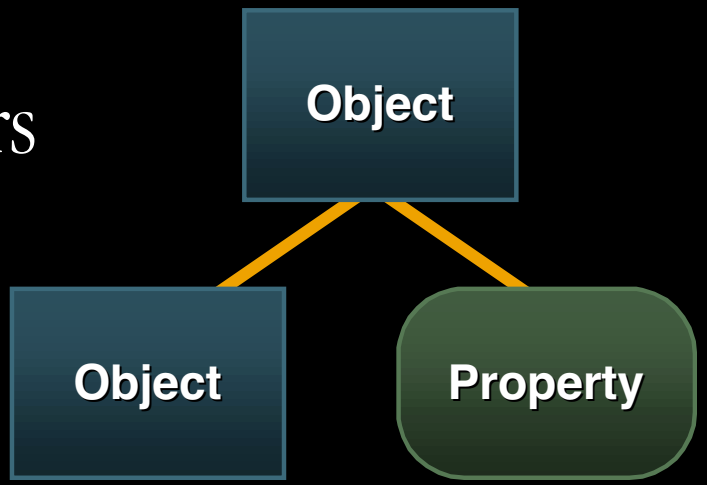


Image Capture Workflow

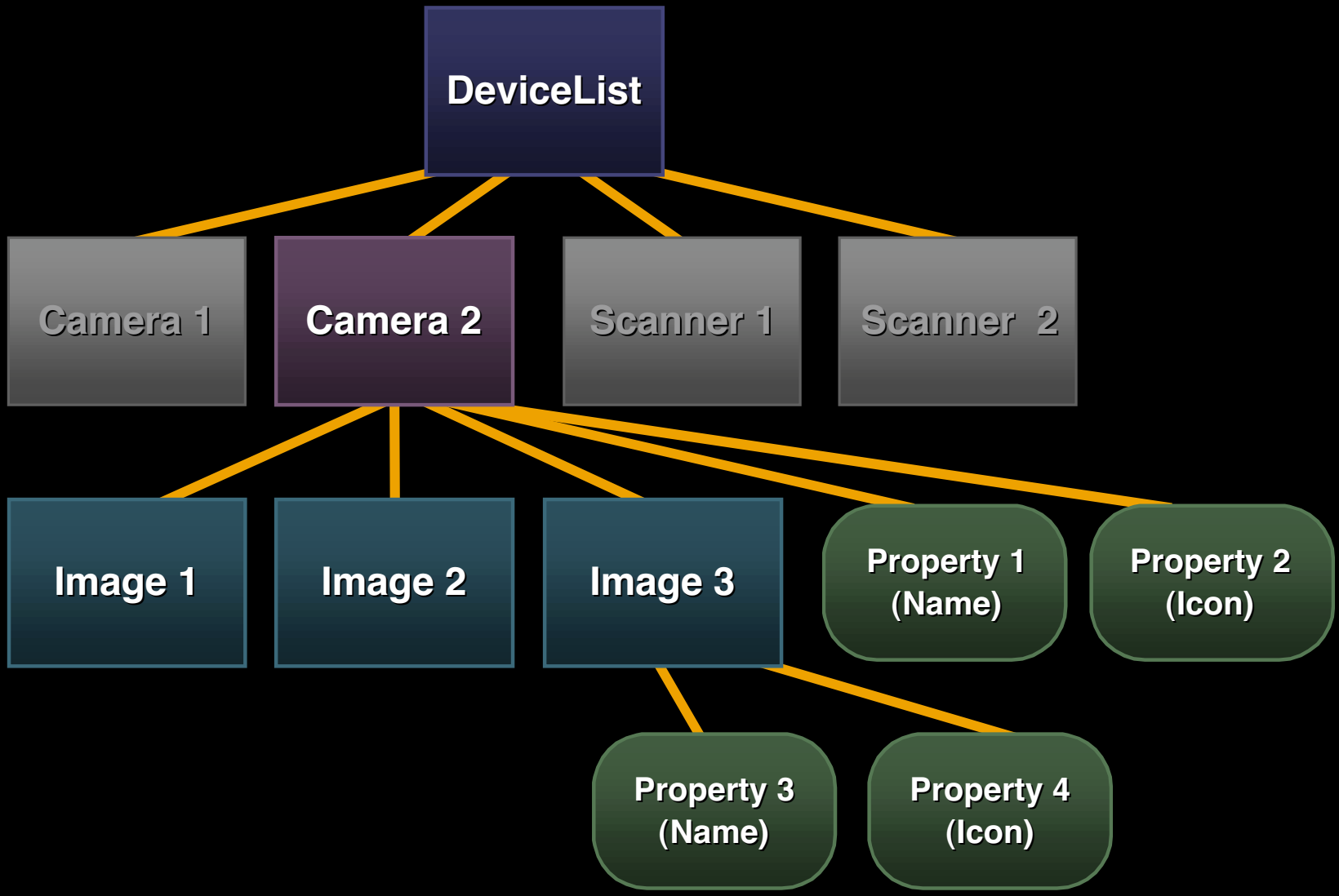


Abstracting the Device

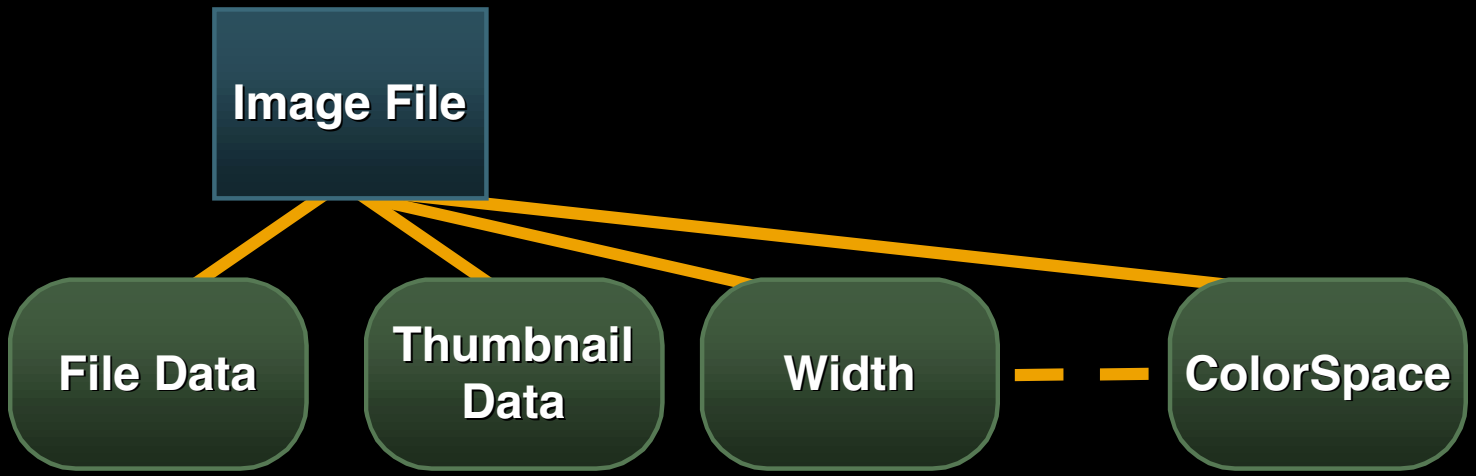
- Objects
  - Type and subtype as identifiers
  - Contain properties, and can also contain other objects
- Properties
  - Type and subtype as identifiers
  - Contain the “real” data



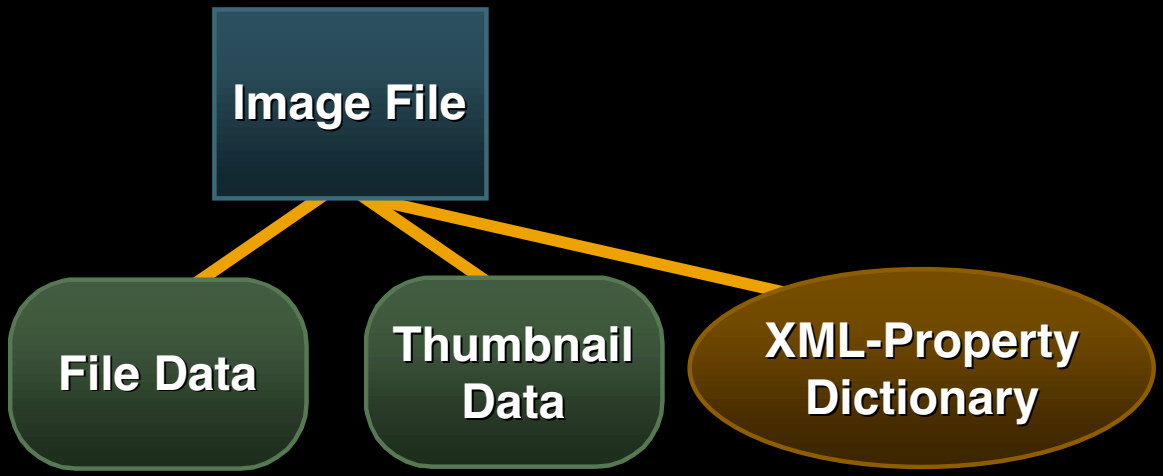
Objects and Properties



Introducing Dictionaries

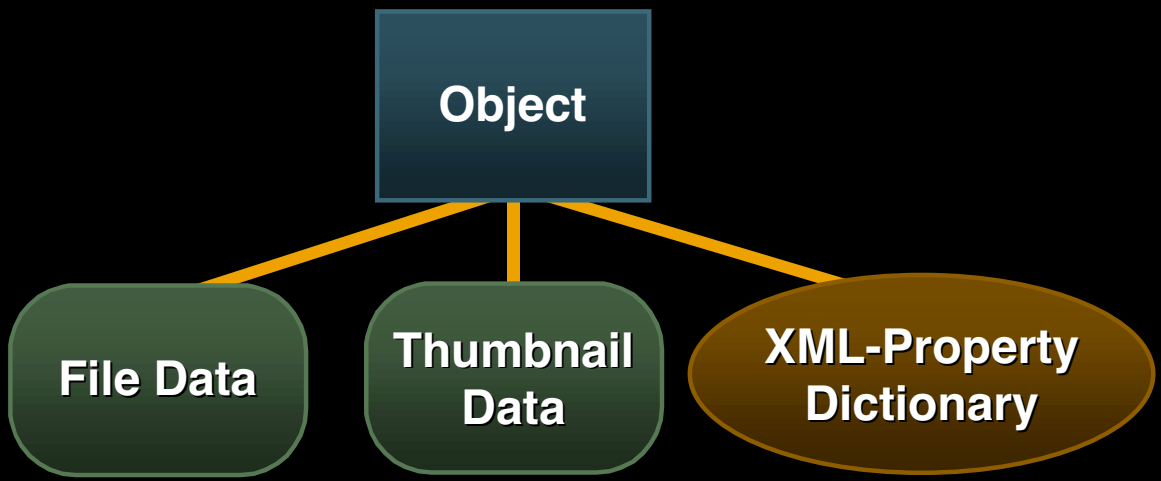


Objects, Properties, and Dictionaries



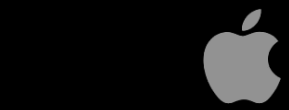
Objects, Properties, and Dictionaries

- Properties contain
  - Image/movie/audio data
  - Thumbnail data
- Property dictionaries contain everything else



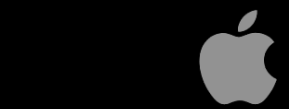
ICACopyObjectPropertyDictionary

- New way to access property data
- Device Objects:
    - Easy way to get device information and basic ‘directory’ information
- Image Objects:
    - Convenient way to get file meta data



Device Object Properties

- Information about the device
  - Name, ICAObject, device capablities, . . .
- Basic information about the device content
  - Flattened out (‘data’ subdictionary)
  - Hierarchical (‘tree’ subdictionary)



Device Object Properties (Cont.)

- ‘data’ sub-directory
  - Fast way to get to the number of images, movies, and audio files
- ‘tree’ sub-directory
  - Convenient way to access all files on a device

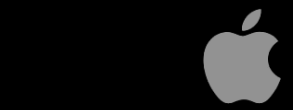


Image Object Properties

- Basic information about the image
  - Name, ICAObject, image data ICAProperty, thumbnail ICAProperty, . . .
- Meta data
  - All meta data, that we can extract

Demo

Image Capture Browser

ICADownloadFile

- Single API to download and post-process images

```
OSErr
ICADownloadFile(   ICADownloadFilePB *  pb,
                   ICACompletion        completion);
```

ICADownloadFile (Cont.)

```
typedef struct ICADownloadFilePB
{
  ICAHeader          header;
  ICAObject          object;
  FSRef *            dirFSRef;
  UInt32             flags;
  OSType             fileType;
  OSType             fileCreator;
  Fixed              rotationAngle;
  FSRef *            fileFSRef;
} ICADownloadFilePB;
```

kDeleteAfterDownload
kCreateCustomIcon
kAddMetaDataToFinderComment
kAdjustCreationDate
kSetFileTypeAndCreator
kEmbedColorSyncProfile
kRotateImage

Demo

Source Code

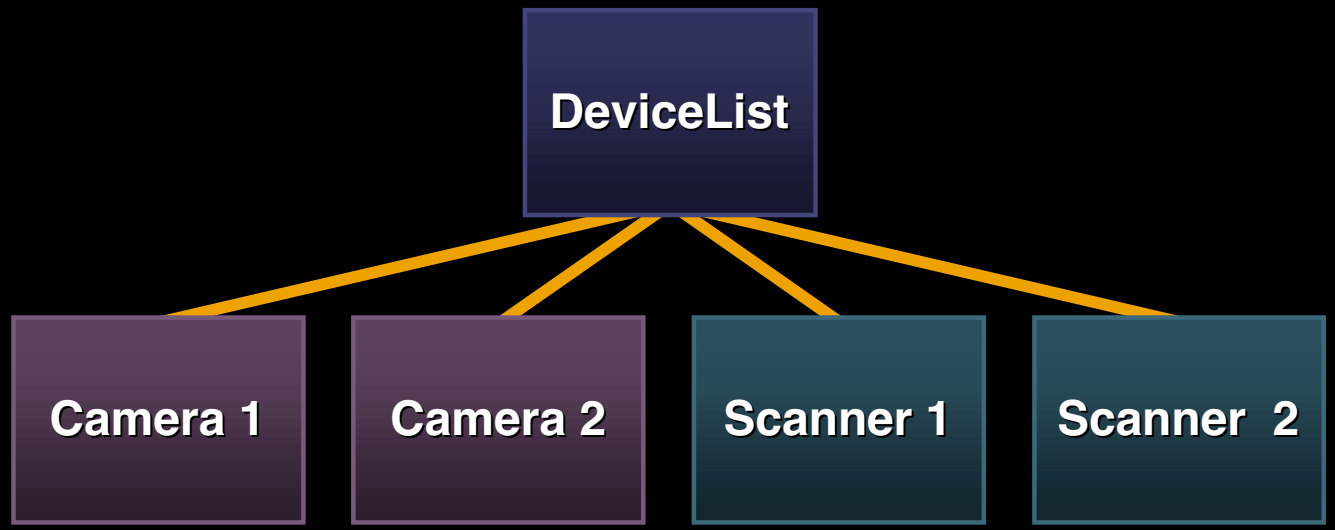
Scanner Support in Image Capture

Scanner Support In Image Capture

- Architecture overview
- Additional APIs for scanners

Architecture Overview

- Image Capture framework supports multiple device types



New APIs for Scanners

- Handling scanner devices is different from Camera devices
    - Session based
    - 7 additional APIs

New APIs for Scanners

- ICAScannerOpenSession

```
OSErr ICAScannerOpenSession (
            ICAScannerOpenSessionPB* pb,
            ICACompletion completion);
```

- ICAScannerCloseSession

```
OSErr ICAScannerCloseSession (
            ICAScannerCloseSessionPB* pb,
            ICACompletion completion);
```

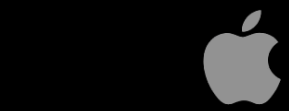
New APIs for Scanners

- ICAScannerInitialize

```
OSErr ICAScannerInitialize (
            ICAScannerOpenSessionPB* pb,
            ICACompletion completion);
```

- ICAScannerStatus

```
OSErr ICAScannerStatus(
            ICAScannerCloseSessionPB* pb,
            ICACompletion completion);
```



New APIs for Scanners

- ICAScannerGetParameters

```
OSErr ICAScannerGetParameters (
            ICAScannerOpenSessionPB* pb,
            ICACompletion completion);
```

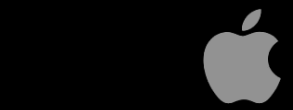
- ICAScannerSetParameters

```
OSErr ICAScannerSetParameters(
            ICAScannerCloseSessionPB* pb,
            ICACompletion completion);
```

New APIs for Scanners

- ICAScannerStart

```
OSErr ICAScannerStart (
            ICAScannerOpenSessionPB* pb,
            ICACompletion completion);
```



Demo

Image Capture Application and Sample Code

What About Scanner Drivers?

- Image Capture supports
  - Image Capture Scanner modules
  - TWAIN data sources

Scanner Device Framework

- Similar to the camera framework
- Contains common code that's used in various scanner modules
- Makes the scanner module development easier

Button Support

- Image Capture framework is supporting device specific buttons such as:
  - Start scan
  - Copy
  - Email
- Support is based on vendor information stored in ‘DeviceInfo.plist’

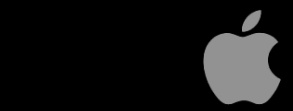
TWAIN Framework

TWAIN Framework

- TWAIN Framework Overview
- Data Sources for Jaguar
- Client Applications

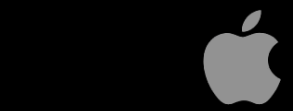
TWAIN Framework Overview

- Established industry wide standard to access imaging devices
- Three key components:
    - Client application
    - Data Source Manager (DSM)
    - Data Source (DS)



TWAIN Framework Overview

- TWAIN is part of Jaguar



TWAIN Framework Overview

- TWAIN is part of Jaguar
- Existing Mac OS X applications using TWAIN will just work—no need to rev

TWAIN Framework Overview

- TWAIN is part of Jaguar
- Existing Mac OS X applications using TWAIN will just work—no need to rev—except . . .

TWAIN Framework Overview

- TWAIN is part of Jaguar
- Existing Mac OS X applications using TWAIN will just work—no need to rev—except . . .

Stop installing the TWAIN shared library…

Mac OS X DSM and DS Locations

- DSM
  - /System/Library/Frameworks/TWAIN.framework
  - For CFM based applications CFM stub lib in
    - /Sytem/Library/CFMSupport/TWAIN Source Manager.Shlb
- DS
  - /System/Library/Image Capture/TWAIN Data Sources

Updated TWAIN DSM for Mac OS X

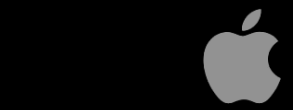
- Supports mach-o based Carbon and Cocoa applications
- Supports mach-o and CFM based Data Sources

Update Your TWAIN DS

- Packaging
- Event handling
- UI-less operation
- DeviceInfo.plist

Mac OS X—DS: Packaging

- Move from CFM based shared libraries to
  - Mach-o or
  - CFM bundles



Mac OS X—DS: Event Handling

- Data Source will not get polled
- Data Source must
  - Implement Carbon Event handler
  - Inform the DSM/client application via a single callback (MSG_XFERREADY, MSG_CLOSEDSREQ, . . .) .

Mac OS X—DS: UI-less Operation

- TWAIN DS must also support UI-less operation (TW_USERINTERFACE.ShowUI false)

Mac OS X—DS: DeviceInfo.plist

- Add a ‘DeviceInfo.plist’ to the DS bundle
- DeviceInfo.plist contains information about supported devices and their capablities

Mac OS X—TWAIN Clients

- Previous version was based on WaitNextEvent
- For client application, this will still work
- Newer Carbon and Cocoa applications have to register notification callbacks

TWAIN Clients—Callbacks

```
DSM_Glue_Entry(&AppIdentity,
                    NULL,
                    DG_CONTROL,
                    DAT_CALLBACK,
                    MSG_REGISTER_CALLBACK,
                    (TW_MEMREF)callback);
```

TWAIN Clients—Callbacks

```
void MyCallback(pTW_IDENTITY pOrigin,   pTW_IDENTITY pDest,
                TW_UINT32    DG,        TW_UINT16    DAT,
                TW_UINT16    MSG,       TW_MEMREF    pData)
{
   switch (MSG)
   {
      case MSG_CLOSEDSREQ:
         TWDisableDS(NULL);
         TWCloseDS();
         break;
      default:
         break;
   }
}
```

Demo

TWAIN Sample Application

Image Capture and TWAIN

- How they can work together (TWAIN Bridge)
- Device arbitration
- Which framework should you use?

TWAIN Bridge

- Image Capture device module that talks to a TWAIN DS
- Monitors device buttons and triggers the correct actions

Device Arbitration

- TWAIN Bridge
  - Gets launched when a device is connected
  - Gets notified when a device button is pressed and launches the appropriate application via the ‘Digital Hub’

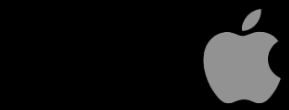
Which Framework Should You Use?

- Depending on your needs
  - Image Capture
    - Easy way to access scanner devices
      - Image Capture modules
      - TWAIN Data Sources
      - …
  - TWAIN
    - More powerful, but steeper learning curve

Changes In Jaguar

Changes In Jaguar

- Digital Hub and application launching
- Extended event notification
- DeviceInfo.plist
- Pass through
- Tidbits



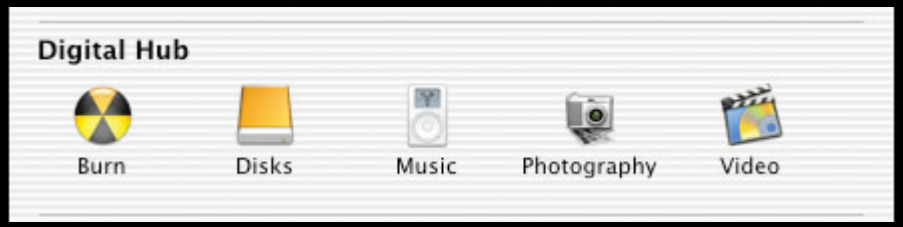
Digital Hub and Application Launching

Mac OS X today:

- Image Capture Application has a popup to select the ‘hot-plug’ application

Digital Hub and Application Launching

- System wide way to specify which application to launch when a device gets connected

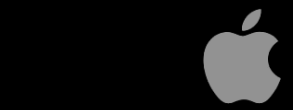


Extended Event Notification

- Client applications can register for event notifications such as
  - Device connected/disconnected
  - Image added/deleted
  - Media card inserted/removed

Extended Event Notification

```
struct ICARegisterEventNotificationPB
{
  ICAHeader             header;
  ICAObject             object;
  OSType                notifyType;
  ICACompletion         notifyProc;
};
```



Extended Event Notification

```
struct ICAExtendedRegisterEventNotificationPB {
    ICAHeader               header;
    ICAObject               object;
    OSType                  extd;
    ICACompletion           notifyProc;

    UInt32                  rawEventType;
    OSType                  eventType;
    OSType                  eventClass;
    UInt32                  eventDataSize;
    ICAEventDataCookie      eventDataCookie;
    ICAObject               deviceObject;
};
```

Extended Event Notification

- Register
  - Set **object** = nil, **extd** = 'extd' to get notified on all new notification events
  - Set **object** = nil, **extd** = 0 to get notified on all notification events
- Unregister
  - Use same parameters a on registration—except pass nil as **notifyProc**

Extended Event Notification

- Register for all events:

```
ICARegisterEventNotificationPB pb;

    memset(&pb, 0, sizeof(ICARegisterEventNotificationPB));
    pb.header.refcon = (UInt32)self;
    pb.object       = 0;
    pb.notifyType       = 0;
    pb.notifyProc       = MyCompletion;
    err = ICARegisterEventNotification(&pb, NULL);
```

Extended Event Notification

- Unregister

```
ICARegisterEventNotificationPB pb;

    memset(&pb, 0, sizeof(ICARegisterEventNotificationPB));
    pb.header.refcon = (UInt32)self;
    pb.object       = 0;
    pb.notifyType       = 0;
    pb.notifyProc       = NULL;
    err = ICARegisterEventNotification(&pb, NULL);
```

DeviceInfo.plist

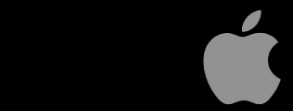
- Each camera module contains a ‘DeviceInfo.plist’ that holds device specific information such as:
  - Device icon
  - ColorSync profile
  - Device class

Pass Through

- Send ‘native’ commands to a camera device
- Easy way to low-level control a device
- Implemented via ICAObjectSendMessage
- Works for PTP, Type5, Type6, and Type7 cameras

Meta Data for Input Devices

- Color issues
- Meta data tags



Color Issues

- Mac OS X is a color managed enviornment
- Weak link in ColorSync chain
    - Images from digital cameras may not contain ColorSync profiles

Image Capture Devices

- Good
  - Use the default profile for input devices
- Better
  - Register a profile with your device
- Best
  - Embed an image specific ColorSync profile

Meta Data

- Preserve meta data in your image processing application

```
err = GraphicsExportGetMetaData(imp, data);
if (err == noErr)
        err = GraphicsExportSetMetaData(exp, data);
```

- Application must check for embedded profile

```
// Copy the profile
        err = GraphicsImportGetColorSyncProfile(imp, &prof);
        if (prof)
          err = GraphicsExportSetColorSyncProfile(exp, prof);
```

CameraCheck

- New tool to test your device with Image Capture
  - Checks camera capabilities
  - Produces a report

Demo

Camera Check

Roadmap

500 Graphics and Imaging Overview	Room A2 **Tue., 10:30am**
501 Quartz 2D and PDF	Room A2 **Tue., 2:00pm**
503 Exploring the Quartz Compositor	Hall 2 **Tue., 3:30pm**
504 OpenGL: Graphics Programmability	Room A2 **Tue., 5:00pm**

Roadmap

505 OpenGL: Integrated Graphics I	Room J **Wed., 9:00am**
506 OpenGL: Integrated Graphics II	Room J **Wed., 10:30am**
109 Darwin Printing	Room J **Wed., 2:00pm**
509 ColorSync and Digital Media	Room C **Wed., 5:00pm**

Roadmap

510 Printing and Mac OS X	Hall 2 **Thurs., 10:30am**
513 OpenGL: Advanced 3D	Room J **Thurs., 3:30pm**
514 OpenGL: Performance and Optimization	Room J **Thurs., 5:00pm**
515 Image Capture Framework	Room C **Fri., 2:00pm**

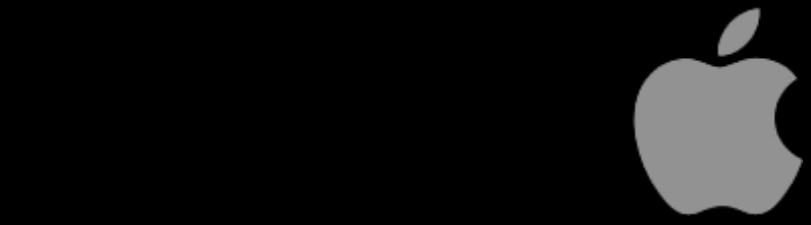
Roadmap

516 Graphics and Imaging Performance Tuning	Hall 2 **Fri., 3:30pm**
FF018 Graphics and Imaging	Room J1 **Fri., 5:00pm**

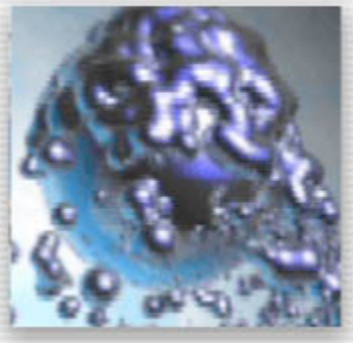
Who to Contact

Travis Brown
Graphics and Imaging Evangelist
Travis@apple.com

http://developer.apple.com/wwdc2002/urls.html



Q&A



Travis Brown
Graphics and Imaging Evangelist
Worldwide Developer Relations

http://developer.apple.com/wwdc2002/urls.html


WWDC2002


WWDC2002


WWDC2002